Problem B. Tree Hull

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

You are given an edge-weighted tree.

Consider a set A which is a subset of vertices of the tree. Initially, A is empty, and we have to process queries which ask to add a vertex to A or remove a vertex from A.

After each query, find the weight of the minimum subtree containing all vertices from A. We define the weight of the subtree as the sum of weights of its edges.

Input

The first line of input contains an integer n: the size of the tree ($1 \leq n \leq 3 \cdot 10^5$).

The next $n - 1$ lines describe edges of the tree. Each edge is described as "u v w": its endpoints and weight ($1 \leq u, v \leq n$, $u \neq v$, $0 \leq w \leq 10^9$). It is guaranteed that the given edges form a tree.

The following line contains an integer q: the number of queries ($1 \leq q \leq 3 \cdot 10^5$).

The next q lines contain queries. Each query is given as "t v", where t is either "+" (add vertex to A) or "–" (remove vertex from A), and v is the number of the vertex ($1 \leq v \leq n$). It is guaranteed that you are never asked to add a vertex which is already in A, or to remove a vertex which is not currently in A.

Output

Print q numbers: the weight of the smallest subtree containing all vertices from A after each query. In case A is empty, print a 0.

Example

standard input	standard output
5	0
1 2 1	1010
2 3 10	1110
3 4 100	110
3 5 1000	111
8	1
+ 2	0
+ 5	0
+ 4	
- 5	
+ 1	
- 4	
- 2	
- 1	